

EXHIBIT 3

Part 2 of 4

Petition for Inter Partes Review of Patent No. 7,953,886

XML is an “extensible” markup language because it allows users to define their own systems of tags and attributes. Ex. 1014 at 10-11. For any particular use of XML, the tags that can be used in an XML document, the type of data they represent, and their hierarchical relationships may be defined in what is known as an “XML schema.” *See id.* at 87-88. A schema acts as a blueprint of the possible combinations of tags and attributes that may be used in an XML document that conforms to a particular application’s syntax and formats. *See id.* at 87-88, 90, 92. Schemas were used to model data structures in the prior art long before the ’886 patent. Clark Decl. ¶ 26; Ex. 1014 at 87-89.

3. Use of XML to Configure Network Equipment

Not long after XML emerged as a versatile and adaptable open standard for data description and transmission, emerging network-technology companies realized its potential use for router configuration. Clark Decl. ¶ 27. By January 2001—over four years before the ’886 patent was applied for—Cisco’s competitor Juniper Networks, Inc. had released an XML-based interface (JUNOScript) for use with its routers, allowing administrators to access and configure the routers using XML commands instead of CLI. *See JUNOScript API Guide and Reference Release 4.3* (5d ed.), Nov. 13, 2001 (Ex. 1015) at 2 (indicating first edition date of January 22, 2001). The JUNOScript API provided a set of XML tags (which it calls JUNOScript tags) that client applications could send to the router. *Id.* at 45. Software

Petition for Inter Partes Review of Patent No. 7,953,886

in the router would process the request, encode the response to the request in JUNOScript tags, and return the result to the client. *Id.*; Clark ¶ 27. The client could thus transmit an XML command to the router and receive a response formatted in XML. Clark Decl. ¶ 27; *see also* Ex. 1015 at 45; *see also* JUNOScript API Guide Release 5.1 (2d ed.), Nov. 6, 2001 (“JUNOScript Guide”) (Ex. 1005).

The shift toward XML, as opposed to CLI, to configure and communicate with network equipment was praised in the industry long before the ’886 patent was filed. Clark Decl. ¶ 28. For example, also in 2001, network- software company Intelliden filed patent applications praising the “user-friendly XML-based interface” of the Juniper Networks routers, while noting that “Cisco™ routers are notoriously difficult to configure” due to their “cumbersome command line interface (CLI).” U.S. Patent No. 8,296,400 (“Gorthy”), issued October 23, 2012 (Ex. 1003) at 1:47-51; *see also* U.S. Patent No. 7,200,548 (“Courtney”), issued April 3, 2007 (Ex. 1002) at 1:47-52. Intelliden acknowledged Cisco’s bind: “If Cisco™ attempted to abandon its CLI in favor of the new user-friendly XML-based interface, many years of development and expertise could be lost.” Ex. 1003 at 1:56-58; *see also* Ex. 1002 at 1:61-63. Anticipating the solution described by the ’886 patent almost four years later, Intelliden filed simultaneous patent applications that would issue as the Gorthy and Courtney patents, disclosing a means for converting XML-based commands to CLI commands, and converting CLI output to XML, for use

Petition for Inter Partes Review of Patent No. 7,953,886

with Cisco routers. *See, e.g.*, Ex. 1003 at 6:7-12, 6:52-59; Ex. 1002 at 2:15-18, 2:32-37.

Others were also developing XML interfaces for existing CLI systems around this time. Clark Decl. ¶ 29. For example, in March 2003, Lee and co-authors published *X-CLI: CLI-Based Management Architecture Using XML* (Ex. 1010), which discloses an API for configuring CLI-based network systems using XML. *Id.* at 3. In addition, U.S. Patent No. 7,155,496 (“Froyd”) (Ex. 1004), filed January 29, 2002, discloses a framework for delivering configuration data in an XML format to and from a system. *Id.* at Abstract. The XML-formatted data, which includes tags containing CLI keywords, is converted into CLI commands that are executed by the system. *Id.* at 8:51-67, 12:44-48.

4. Identifying, Extracting, and Arranging Components of Commands Received in an XML Format

A person of ordinary skill in the art (“POSA”) at the time of the ’886 patent application would have understood that a CLI command in XML format could readily be transformed by using, for example, the well-known text-manipulation tools of Unix, such as “sed,” “grep,” and “awk.” Clark Decl. ¶ 30; *see also* Ex. 1009 at Chapter 4. A POSA would further have understood that these and other techniques for identifying, extracting, and arranging could readily be applied to XML. Clark Decl. ¶ 31. Indeed, transforming XML into a different form was a well-known use for XML long before the ’886 patent was filed. *Id.* As *Learning*

Petition for Inter Partes Review of Patent No. 7,953,886

XML explained in 2003, “[t]ransformation requires two things: the source document and a transformation stylesheet. The stylesheet is a recipe for how to ‘cook’ the XML and arrive at a desired result. . . . [Transformation] can be used to turn an XML document into just about any form you can imagine.” Ex. 1014 at 50-51.

V. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art is presumed to be aware of all pertinent art, uses conventional wisdom in the art, and is a person of ordinary creativity. With respect to the ’886 patent, a POSA would typically have at least a bachelor’s degree in computer science and 3-5 years of experience in systems development. Clark Decl. ¶ 16.

VI. CLAIM CONSTRUCTION

The Board gives claims their broadest reasonable interpretation (“BRI”) consistent with the specification. 37 C.F.R. § 42.100(b); *In re Cuozzo Speed Technologies, LLC*, 793 F.3d 1268, 1279 (Fed. Cir. 2015).¹ Such a construction must

¹ Other forums, such as district courts, apply a different standard. See *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*). Any interpretation of the challenged claims in this petition, either explicit or implicit, does not reflect Arista’s interpretation under a standard other than the BRI. In particular, Arista reserves the right to argue for different constructions under the *Phillips* standard in the related district-court litigation.

Petition for Inter Partes Review of Patent No. 7,953,886

reasonably reflect the plain language and disclosure of the patent “as [they] would be interpreted by one of ordinary skill in the art.” *In re Suitco Surface Inc.*, 603 F.3d 1255, 1259 (Fed. Cir. 2010).

As relevant here, the Board should construe the claim terms as follows:

A. “command line interface (CLI) parser”

Proposed Construction: “a subsystem of the routing system capable of receiving input commands, translating those commands into CLI commands, and parsing the received and translated CLI commands.”

The claims of the ’886 patent recite that the CLI parser receives commands in an XML format and translates them to a CLI format. For example, they state that the “command line interface (CLI) parser” “receiv[es] . . . an input command” configured in an XML format having a CLI syntax, and “tranlat[es] the input command from the XML format having the CLI syntax into a CLI command” Ex. 1001 at 7:38-47; *see also id.* at Fig. 1 (showing IOS/CLI Parser 110) and 3:20-22; Clark Decl. ¶¶ 34-35.

The received command statements can be in a CLI format or an XML format—that is, they can be “formatted in accordance with the CLI rules and behaviors expected by IOS/CLI Parser 110, or in accordance with an XML schema of the CLI rules and behaviors.” Ex. 1001 at 3:23-26. “Non-CLI command statements” (which would include input commands in an XML format) “must . . . be translated

Petition for Inter Partes Review of Patent No. 7,953,886

into CLI statements that Command Module 130 can interpret.” *Id.* at 3:34-36. The “command line interface (CLI) parser” performs this translation, as shown in Figure 2, which is “a flowchart of a method for *receiving* and *translating* data using an internetwork operating system (IOS) *command line interface (CLI) parser subsystem* of a routing system, in accordance with one embodiment of the present invention.” *Id.* at 1:44-48 (emphasis added). Once the CLI commands are received by the CLI parser, either directly or after being translated from an XML format, “the command statements are parsed according to instructions programmed into IOS/CLI Parser 110.” *Id.* at 3:29-31.

B. “parsing the output message to identify at least one CLI token”

Proposed Construction: “analyzing the output message to extract at least one unit of CLI characters in a sequence.”

Cisco Systems, Inc. (“Cisco”), the owner of the ’886 patent, has proposed the above construction for “parsing the output message to identify at least one CLI token” in the related litigation. Cisco’s Preliminary Claim Construction Disclosure, *Cisco Systems, Inc. v. Arista Networks, Inc.*, No. 5:14-cv-05344-BLF (N.D. Cal. Aug. 24, 2015) (Ex. 1012) at 26. Arista agrees that Cisco’s construction would be the broadest reasonable interpretation of this term. The term “parsing” generally refers to a “process by which an input string is analyzed using a grammar to determine if the input string satisfies the rules of the grammar.” Dictionary of Computer

Petition for Inter Partes Review of Patent No. 7,953,886

Science, Engineering, and Technology (2010) (Ex. 1017) at 4; *see also* Ex. 1012 at 26. The term “token” generally refers to “[a] distinguishable unit in a sequence of characters.” *See* McGraw-Hill Dictionary of Scientific and Technical Terms, 6th ed. (2003) (Ex. 1013) at 6 (definition 1); Clark Decl. ¶ 26.

VII. SUMMARY OF THE PRIOR ART FORMING THE BASIS OF THIS PETITION

A. Courtney

U.S. Patent No. 7,200,548 to Courtney, entitled “System and Method for Modeling a Network Device’s Configuration,” issued from an application filed on August 29, 2001. Courtney is accordingly prior art under 35 U.S.C. § 102(e). Moreover, the Courtney application was published on March 6, 2003, making the published application prior art under 35 U.S.C. § 102(b).

Courtney discloses a “system and method for modeling the configuration of a network device” that may include “a CLI-to-XML converter.” Ex. 1002 at 2:32-37; *see also* Clark Decl. ¶¶ 38-39. As with the ’886 patent, one reason for using the Courtney invention would be to “model a network device’s configuration” by “converting it into a standard-format configuration such as an XML document or a DOM.” Ex. 1002 at 2:40-45. This was beneficial to users because “instead of being forced to manipulate a difficult CLI-based configuration format, . . . system administrators can use the standard-format configuration to interact with the target network device.” *Id.* at 2:48-51. In order to perform the translation to XML, the sys-

Petition for Inter Partes Review of Patent No. 7,953,886

tem could “generate[] an XML representation of each native-format command in the network device’s configuration by associating each command with the schema, or its hash representation.” *Id.* at 3:23-27; *see also id.* at 6:50-53 (“[T]he XML converter 235, using the appropriate schema, generates an XML document containing an XML representation of the network device’s configuration.”).

The Courtney system uses schema information to take the native-format configuration and “assemble the XML-based command and write it to the corresponding XML document.” *Id.* at 7:31-36. That process is repeated for each command in the device’s native-format configuration until “all native-format commands have been converted.” *Id.* at 7:37-48.

B. Gorthy

U.S. Patent No. 8,296,400 to Gorthy *et al.*, entitled “System and Method for Generating a Configuration Schema,” issued from an application filed on August 29, 2001. Gorthy is accordingly prior art under 35 U.S.C. § 102(e). Moreover, the Gorthy application—which is incorporated by reference into Courtney—was published on March 13, 2003, making the published application prior art under 35 U.S.C. § 102(b).

Gorthy discloses a configuration schema that is generated by collecting the commands used with a Cisco router. Ex. 1003 at 2:30-59. That configuration schema can then “be used to generate commands.” *Id.* at 2:63-65; *see also id.* at

Petition for Inter Partes Review of Patent No. 7,953,886

3:9-11 (“[T]he schema can be used to generate CLI commands from, for example, XML-based commands.”). Thus, “[w]hen given a command in XML format, the command information in the configuration schema can be used to reformat the XML-based command into a proper CLI format.” *Id.* at 3:13-16. After reformatting, the command can be sent to the router, allowing a system administrator to configure routers without knowing the details of the CLI for the router. *See id.* at 3:16-19; *see also* Clark Decl. ¶ 41.

The invention of Gorthy allows an “XML-based command [to] be passed to the converter 235 which converts the XML-based command to a CLI-based command using the XML schema.” Ex. 1003 at 6:9-11. Thereafter, the “CLI-based command, not the XML command, can then be passed to the configuration storage module 145 where it is integrated into the configuration of the router.” *Id.* at 6:11-14.

C. Froyd

U.S. Patent No. 7,155,496 to Froyd (Ex. 1004), entitled “Configuration Management Utilizing Generalized Markup Language,” issued from an application filed on January 29, 2002, as a continuation-in-part of an application filed on May 15, 2001. Froyd is accordingly prior art under 35 U.S.C. § 102(e). Moreover, the Froyd application was published on June 19, 2003, making the published application prior art under 35 U.S.C. § 102(b).

Petition for Inter Partes Review of Patent No. 7,953,886

Froyd discloses a framework for delivering data to and from a system. Clark Decl. ¶ 44. It recognizes that data in an XML format is oftentimes more versatile for users than data kept in a native format such as that used by a CLI. Ex. 1004 at 15:60-63 (“Placing the statistic information into the XML format also allows the statistic information to be used by other devices capable of processing data in the XML format.”). Accordingly, Froyd proposes a system capable of sending and receiving information about how a system is or should be configured and that converts the information between XML and CLI formats. Clark Decl. ¶ 45. For example, “[i]n one embodiment, the configuration file is saved in XML format. When a restore function is invoked, the configuration file 818 is processed into CLI command lines. The CLI command lines are fed back to the CLI 805.” Ex. 1004 at 12:44-48.

The Froyd system also recognizes the need to make the commands that it uses consistent with the language used by the router in question. Clark Decl. ¶ 46. “XML can be used to create a new set of commands that describe commands associated with a Lucent router such that the new set of commands is consistent with commands associated with routers from Cisco.” Ex. 1004 at 11:20-24.

Petition for Inter Partes Review of Patent No. 7,953,886

D. JUNOScript API Guide

JUNOScript API Guide, Release 5.1 (2d ed.) (“JUNOScript Guide”) (Ex. 1005), was published on November 6, 2001, by Juniper Networks, Inc. The JUNOScript Guide is accordingly prior art under 35 U.S.C. § 102(b).

The JUNOScript Guide describes the use of the JUNOScript application programming interface (API) to configure or request information from software running on Juniper Networks routers. Ex. 1005 at 9; *see also* Clark Decl. ¶ 48. The JUNOScript API uses XML tags that describe router components to perform these operations. Ex. 1005 at 9. The XML tags correspond to CLI statements used to configure or request information from the router. Clark Decl. ¶ 48; Ex. 1005 at 32 (“Configuration requests correspond to the JUNOS CLI configuration statements described in each of the JUNOS Internet software configuration guides. The JUNOScript API defines a tag for every container and leaf statement in the JUNOS configuration hierarchy.”).

The JUNOScript Guide describes the benefit of formatting inputs to and outputs from the router in XML tags, noting that the tags “make it straightforward for client applications that request information from a router to parse the output and find specific information.” Ex. 1005 at 17. The Guide illustrates this by showing how tokens of an output message received in response to a CLI command are translated into JUNOScript XML values. *Id.*; *see also id.* at 37 (“To display the

Petition for Inter Partes Review of Patent No. 7,953,886

output from a JUNOS CLI command as JUNOScript tags rather than the default formatted ASCII, pipe the command to the **display xml** command.”); Clark Decl. ¶ 49.

E. Motivation to Combine

A POSA would have every motivation to combine Courtney and Gorthy—indeed, it may be unnecessary to combine them, as Courtney expressly incorporates by reference the application that matured into Gorthy. Ex. 1002 at 3:7-11 (incorporating by reference U.S. Patent Application No. 09/942,834). The applications that matured into Gorthy and Courtney were filed the same day (in fact, they bear consecutive serial numbers) and describe two facets of a system from a company known as Intelliden (Intelliden was acquired by IBM in 2010, before Gorthy issued, thus explaining the different assignees shown on the face of the two patents).

A POSA would also be motivated to combine Courtney and Gorthy because their inventions are complementary to one another. Clark Decl. ¶ 51. Courtney explains how to use the conversion of CLI to XML in order to allow configuration information sent from a router to be easily understood. *Id.* Gorthy provides information about the use of the flip side of this process—how to use the conversion of XML to CLI to allow routers to be configured in a simplified manner. *Id.*

Petition for Inter Partes Review of Patent No. 7,953,886

Likewise, one of skill in the art would have been motivated to combine Froyd with either or both Courtney and Gorthy because each is concerned with the use of translations to and from XML in order to provide users with an easy to use standard format. *Id.* at ¶ 52; *see, e.g.*, Ex. 1002 at 2:48-51 (“That is, instead of being forced to manipulate a difficult CLI-based configuration format, or other format system administrators can use the standard-format configuration to interact with the target network device.”); Ex. 1003 at 3:5-8 (“Once the CLI-based command has been converted to an XML format, the XML format of the command can be easily passed between various computers and system administrators in a highly readable, standardized format.”); Ex. 1004 at 15:57-63 (“The statistic information is returned to the user in the XML format. This allows the statistic information to be displayed and presented to the user through the browser. Placing the statistic information into the XML format also allows the statistic information to be used by other devices capable of processing data in the XML format.”).

Moreover, all three of the references expressly discuss the potential for use with Cisco routers specifically. *See, e.g.*, Ex. 1002 at 2:15-18 (“Accordingly, a system and method are needed that will allow manufacturers, like Cisco™, to create user-friendly interfaces for both next-generation and existing devices.”); Ex. 1003 at 4:39-41 (“The illustrated method can be used, for example, to generate an XML schema from the CLI commands associated with a Cisco™ router.”); Ex. 1004 at

Petition for Inter Partes Review of Patent No. 7,953,886

11:20-24 (“XML can be used to create a new set of commands that describe commands associated with a Lucent router such that the new set of command is consistent with commands associated with routers from Cisco.”); *see also* Clark Decl. ¶ 53.

A POSA would further have been motivated to combine the JUNOScript Guide with Gorthy, Courtney, and Froyd because each reference relates to using XML-tagged commands to configure and request information from routers. Clark Decl. ¶ 54. Both Gorthy and Courtney discuss the XML-based interface of Juniper Networks routers (Ex. 1003 at 1:47-51; Ex. 1002 at 1:48-51), and Gorthy specifically illustrates that configuration schemas according to the invention could be generated for Juniper Networks routers (*see* Ex. 1003 at Fig. 4). Those references’ praise of such XML-based interfaces would have directed a person of ordinary skill in the art to documentation regarding Juniper Networks routers, such as the JUNOScript Guide. Clark Decl. ¶ 54. Moreover, these references are directed to the same field of endeavor—frameworks for configuring networking equipment using XML—and teach similar methods of using XML-based commands with a CLI syntax. *Id.*; *see, e.g.*, Ex. 1004 at Abstract, Fig. 7; Ex. 1005 at 15-16.

A POSA would further be motivated to combine the teachings of the JUNOScript Guide with Gorthy, Courtney, and Froyd because the inventions complement each other. Clark Decl. ¶ 55. Courtney, which incorporates Gorthy by ref-

Petition for Inter Partes Review of Patent No. 7,953,886

erence, discloses a CLI-to-XML converter for use with a router (Ex. 1002 at Abstract), and the JUNOScript Guide teaches outputting the results of a CLI command to XML-formatted tags (*see, e.g.*, Ex. 1005 at 37). A POSA would understand that the JUNOScript Guide discloses possible implementations of CLI-to-XML conversion that could be used consistent with the system disclosed in Gorthy and Courtney. Clark Decl. ¶ 55. Likewise, Froyd discloses an XML tagging convention for CLI commands that one of ordinary skill in the art would recognize could be used in conjunction with the system of Gorthy/Courtney and the JUNOScript Guide. *Id.*

VIII. IDENTIFICATION OF CHALLENGE (37 C.F.R. § 42.104(b))

Petitioner requests *inter partes* review and cancellation of claims 1-10 of the '886 patent as being obvious over Courtney in light of Gorthy, Froyd, and the JUNOScript Guide.

A. Claim 1

1. [1A] “A method comprising:”

Gorthy in combination with Courtney, Froyd, and the JUNOScript Guide discloses a method with the steps set forth in the elements below. *See* claim elements 1B-1E, below; *see also* Clark Decl. ¶ 57.

Petition for Inter Partes Review of Patent No. 7,953,886

2. [1B.1] “receiving, with a command line interface (CLI) parser, an input command configured to request an operation be performed by a routing system,”

Gorthy discloses receiving, with a command line interface (CLI) parser, an input command configured to request an operation be performed by a routing system. *See Clark Decl.* ¶¶ 58-60. For example, Gorthy discloses that the “converter 235’ initially receives an XML-based configuration command.” Ex. 1003 at 6:43-44; *see also id.* at 6:9-11 (“That XML-based command can be passed to the converter 235 which converts the XML-based command to a CLI-based command using the XML schema.”).

The XML-based input commands disclosed by Gorthy are configured to request an operation be performed by a routing system. *See Clark Decl.* ¶ 59. Gorthy indicates that “a system administrator 125 can reconfigure such a router using XML-based commands . . .” Ex. 1003 at 6:4-5. The exemplary schema of Appendix B, which is used to generate certain of the exemplary CLI commands in Appendix A, for example, defines certain “service” configuration commands that would request an operation be performed by a routing system. *Clark Decl.* ¶ 59. Moreover, Gorthy explains that “[o]nce reformatted into a CLI format, the command can be pushed out to the appropriate router. Thus, a system administrator could configure such a router without knowing the specifics of the CLI.” Ex. 1003 at 3:16-19; *see also id.* at 4:33-35 (“[N]ew configuration commands are provided

Petition for Inter Partes Review of Patent No. 7,953,886

to the router 120 through the CLI.”). A POSA would understand that a command that is “pushed out to the appropriate router” is configured to request an operation be performed by a routing system, for example, enabling the exemplary “service” features disclosed in Appendices A and B. Clark Decl. ¶ 59.

The input command described above is received with a “command line interface (CLI) parser.” *Id.* at 60. Gorthy discloses that its routing system includes a converter subsystem. Ex. 1003 at Fig. 5; *see also id.* at 3:41-42, 5:64-67. Gorthy explains that the converter 235 receives XML-based configuration commands and “converts the XML-based command to a CLI-based command” *Id.* at 6:9-11; *see also id.* at 6:43-44. After translating the input command to a CLI command, the converter “verifies the correctness and validity of that command, and provides it to the router 120” *Id.* at 6:48-51. Because it receives an XML input command, translates the XML input command to a CLI command, and parses the resulting CLI command, the converter of Gorthy discloses the claimed “command line interface (CLI) parser.” Clark Decl. ¶ 60.

3. [1B.2] “wherein the input command is configured in an extensible markup language (XML) format having a CLI syntax with CLI keywords sequenced according to configuration rules for CLI commands;”

Gorthy discloses that the input command is configured in an extensible markup language (XML) format having a CLI syntax with CLI keywords sequenced according to configuration rules for CLI commands. Clark Decl. ¶¶ 61-